COMPUTER NOTES
©mits
Vol. 2 Issue 7
December, 1976

Table of Contents

LETTER TO THE EDITOR

Dear Ed.,

While programming my Altair computer, I detected a potential problem with the 8800 SIO-B (ACR) I/O card which may be of interest to other users. Although there is a solution, it's not entirely satisfactory. I hope someone else can supply a better one.

The problem is that an input program written at one location will work while at another it won't. The program and the address at which the program won't work is listed as follows:

001	IN	333
002	Stat chan	002
003	RRC	017
004	JC	332
005	L	001
006	H	any page
007	IN	333
010	Data chan	003
011	ETC	

The problem occurs if the program is written in a location such that the low order address of the instruction (RRC) following the input instruction (2nd byte) is equivalent to the data channel address which is 003 in this case. The result is that if a string of characters is being input, about 1 in 20 will be lost.

During machine cycle 1 (M1) of the input instruction (see Figure 1), the first byte (333) is read into the processor and during M2 the address of the device (002-status channel) is read in. During M3 the processor outputs the device number (as an address) from which data is also to be read. The processor reads in the data during T3 of M3.

During each of the machine cycles of any instruction, the processor outputs a status word which is latched into the status register (on the CPU card) by the coincidence of Ø1 and sync. The status word is used to control memory and I/O cards. In this case the status

Bits and Pieces

BY: SONDRA KOPPENHEFFER

FATAL ACCIDENT DAMPENS THE CHRISTMAS SPIRIT

Sending Christmas packages through the mail is a part of the holiday spirit. However, packages have been known to experience quite a lot of trauma before reaching their intended destination. Let me contrive an example to convey the message of this article.

Your disk, never having much of a Christmas spirit, has decided to take a vacation. No amount of intuition or technical ability will convince that marvel of technology to work. Giving up on the disk, for more enjoyable holiday activities, you finally decide to give MITS a try at repairing the disk.

. . . About a week later, you receive a phone call from MITS, explaining that your disk had been involved in a fatal accident. Upon arrival at MITS, the disk had been severely damaged, and the $200.00 for which you had insured it, wasn't nearly enough to cover the doctor's bills. Your bereavement is turned into shock when you are told that you will have to pay a $350.00 bill for repair of the disk.

The gentleman in this story had to learn about our postal system the hard way. But this type of misfortune should be a lesson to all Altair owners who send their units back to MITS for repair. Always insure your system for at least the amount that you originally paid for it. Even if the case is the only part of your system damaged in transit, you will still be billed $170.00 for repairing the case.

So remember to send us your unit by registered mail and make sure it's insured for the full amount.

UPDATE SERVICE TAKES A SEAT IN HISTORY

The advertisement for Computer Notes Review, Vol. 1, has generated many inquiries concerning our update service. However, this service was discontinued six months ago. We feel that any benefits you would have received through the Update Service, are now directly available in Computer Notes.

CUSTOMERS SUBMIT MISSING PAPER REPORT - MASSIVE SEARCH BEGUN AT MITS

This paragraph is dedicated to all those customers who have recently purchased an Altair computer through one of our dealers. We've received many calls and letters asking why new customers have not started receiving Computer Notes. We depend upon our dealers to submit the names of all those persons who purchase an Altair through their store. This procedure generally takes three weeks to one month. Once in a while a dealer inadvertently misses a name and we are therefore unable to place that customer's name on our mailing list. If you find yourself in the situation which I just described, please contact the dealer from whom you purchased your Altair. Following up on this procedure will help limit any duplication in your subscription.

COMPLETE RECORD OF PURCHASES NEEDED

If you have ordered systems through both MITS and one of our dealers, you might find it to your best interest to send us copies of all of your dealer purchases so that we might update your file. If we have complete records for all Altair products which you have purchased, you can depend on much more efficient and rapid service.

Continued on Page Seven

On The Cover

This picture of Ron Roberts, president of the Altair Software Distribution Company, is just one example of digital art produced on an Altair 8800b computer.

This system was developed by The Computer Systemcenter in Atlanta and uses a Panasonic video camera, a centronics printer and a commercially available digitizer, in addition to an 8800b with special interfaces and custom software.

The system works by assessing the relative amount of gray in a TV picture, setting a character on the printer that approximately corresponds to that perceived amount of gray and finally, printing the picture.

COMPUTER NOTES

Editor Andrea Lewis
Asst. Editor Linda Blocki
Production Tom Antreasian, Al McCahon, Steve Wedeen, Grace Brown
Contributors Tommy Staten, Gene Dial, Charles Olsen, Douglas L. Jones, Mike Hunter, Bob Matthews, Sondra Koppenheffer, Jim Wiggins, Rich Haber, Bruce Fowler

WACC-II SLATED FOR MAY

By Charles Olsen

The second annual World Altair Computer Convention(WACC-II), scheduled for May 18-21 at the Albuquerque Convention Center, promises to be even more spectacular than last year's, with the largest collection of Altair equipment, dealers and computer experts anywhere in the country.

Attendance is estimated at more than 1,000 computer enthusiasts, which is well over the unexpected 750 people who gathered at last year's WACC.

The four-day convention will be geared to both the hobbyist and small businesses but will also include displays of many other special applications for large businesses, industry, education and the home.

All of our Altair dealers will be at the convention to display various systems and answer any questions.

A banquet will be held on May 20, and the last day of the convention will be reserved for various special events, such as awarding prizes and conducting tours of the Albuquerque-Santa Fe areas.

Lodging will be available at the Albuquerque Inn, which is connected by an underground walk-way to the Convention Center next door. Only those people who confirm their reservations early will get their choice of rooms at the Albuquerque Inn. Additional accommodations will be available at the local Sheraton and Hilton hotels, only a 10-minute drive from downtown Albuquerque.

Details concerning the WACC-II schedule, seminars, dealer contacts, prizes and tickets for the banquet will appear in next month's issue of Computer Notes as well as other computer magazines.

If you have any further questions about WACC-II, please feel free to contact me.

Charles Olsen
Director of PR
MITS, Inc.

Book Review

Practical Microcomputer Programming
The Intel 8080

By: Mike Hunter

Although most microcomputer users usually prefer to use a high level language, such as BASIC, when writing programs, they are often either compelled by necessity or motivated by interest to use machine code or assembler instead. Learning how to manipulate the ones and zeroes as well as the various mnemonics can be confusing for anyone. But it's particularly devastating to the programmer who has only one or two kilobytes of memory and thus has no other choice but to use machine code.

The literature available from the chip manufacturers sometimes provides only brief explanations of machine code functions. For the novice microcomputer user, such descriptions are often as cryptic as the mnemonics. One excellent solution to this problem is Weller, Schatzel and Nice's brand new book, Practical Microcomputer Programming The Intel 8080.

The book provides the novice with over 300 pages of all the information necessary to begin programming a microcomputer. Although the book is aimed at beginners, it's still a very helpful, detailed source of reference material for programming experts.

The 18 chapters of the book cover a wide variety of topics, including binary addition, how to use the stack pointer and how to interface the 8080 to complex peripherals. Each chapter begins with a discussion of a basic concept followed by instructions on how to implement the concept with the 8080 machine code.

The book is based on a configuration consisting of a Teletype™ and an Altair 8800, which was assembled by one of the authors. The authors wrote their own unique assembly program to be used as the main example program throughout the book.

Many instructive texts are often boring as well as difficult to understand. However, Practical Microcomputer Programming is not only interesting but is written in a style that even beginners can follow. The authors help alleviate the drudgery and confusion of working through examples by using such names as Tom, Dick and Harry as variables rather than the traditional X and Y.

The final chapters of the book discuss such questions as, "How do I interface my 8080 microcomputer to an analog device, or terminal, and how can it be used in a real time mode?"

The last chapter deals appropriately enough with the final step in the programming process--debugging. This chapter describes many procedures for discovering the source of an error.

In general I'd recommend this book as a valuable source of programming information for any 8080 microcomputer user.

Practical Microcomputer Programming--The Intel 8080 is available for $21.95 from:

Northern Technical Books
Box 62
Evanston, Ill. 60204

New Club Formed

State University of New York at Stony Brook recently formed the Home Brew Computer Club to promote informal discussions of a variety of both hardware and software topics. Most members have their own systems (8080 and 6800) to discuss and demonstrate. The club also has access to a number of simulators and cross-assemblers on the university computer. For further information, contact:

Ludwig Broun
Professor of Engineering
College of Engineering and Applied Sciences
State University of New York at Stony Brook
Stony Brook, NY 11794

December Software Contest

The Altair User Group Software Library is in the process of being transferred to the same facility that houses the new Altair Software Distribution Company. Because we are in the middle of this transition and because there were only a few entries for December, no contest winners were chosen this month. All entries received in December will be included in the January contest.

Check the January issue of CN for further new developments in the Altair Users Group Software Library.

Case Study:

Monitoring Lumber Production with an Altair Computer

By Tommy Staten

Mr. Staten is the mill manager of H.C. Hodges Lumber Company in Panama City, Florida.

Since the H. C. Hodges Lumber Company first opened, we've had difficulty keeping track of our production. The lumber was being tallied daily as it was stacked, but at the end of each month, there was a considerable discrepancy between what was counted and what was inventoried. Obviously, this caused quite a few problems. So I devised a method of using an Altair computer to monitor production.

I chose the Altair 8800 microcomputer because I am convinced that no other manufacturer matches the MITS Altair system in either hardware or software support. Price was also a factor. Although the Altair is not the cheapest computer on the market, it has proven it was certainly worth the cost by being a very reliable and efficient system for our lumber company.

The following description is only one example of how an Altair system can be used in an industrial application. The possible uses of such a microcomputer system are almost limitless, not only within the lumber industry but in any operation where production control and cost effective monitoring systems are required.

In the past three years more and more manufacturers have begun to incorporate solid state electronic circuitry in their system controls rather than using the usual magnetic relays, mechanical switches, limit swiches, etc. These methods were fine for the actual operation of a machine, but they proved to be very clumsy for automatic control of equipment.

The advent of solid state electronics in system controls has resulted in increased production that's less expensive and more efficient. With automatic controls, "down time" or lost production is also minimal.

The lumber industry is a prime example of how solid state electronics in system controls have been effectively used to accurately monitor daily production while saving both time and labor costs. Four years ago the average lumber mill used 15 to 20 men to sort and stack rough cut lumber as it came out of the mill. The usual procedure utilized a 150-foot lumber conveyor chain that moved the lumber from the cutting saw area. As the lumber came out, men stationed on both sides of the conveyor would pull, sort and stack the lumber according to size. Today, that long conveyor still exists, but instead of 15 men, only one--the control operator--is needed to get the job done. This change was made possible by using a new dropsorter built by Harvey Engineering, Little Rock, Arkansas.

The dropsorter is a low voltage, solid state binary logic control system that uses primarily CMOS ICs and some TTL. Although the limit switches were not entirely eliminated, their use was limited strictly to generating timing and measurement signals. The following description explains how the system works.

Dropsorter:

Sequence of Operation

The dropsorter control logic system is composed primarily of five printed circuit boards interconnected with 26 conductor ribbon cables. Control system input signals of 24 VDC are supplied to the I/O board from limit switch contact closures and/or operator control push buttons.

When a piece of lumber passes through the measuring section of the sorter infeed, certain limit switches are actuated, and width, length and thickness signals are coupled from the I/O board to the measuring board where they are connected to 12 VDC binary logic. The signals are then coupled to the display board, which displays the actual board dimensions. A proper dimensional display indicates that the external switches, I/O board and measuring board are functioning properly.

At the same time the input signal is displayed, the binary coded signal is also fed to the program coding board. The function of the coding board is to assign a two-bit binary bay address to each major lumber dimension(represented by the binary coded signal).

Any lumber dimension which does not have a programmed lumber bay address is automatically dropped in a designated drop-out bay.

The two-bit binary bay address is coupled from the program board to the memory or shift board, where the binary bay address can be seen from the time a piece of lumber is measured until the reset limit switch resets the system for the next measurement. The two-bit binary address also enters a multichannel shift register, which shifts one position for every revolution of the dropsorter head shaft. The number of shifts required for a board to travel to a specific bay must be predetermined, and the memory or shift board should be preprogrammed for the proper number of shifts before sending an output signal to a designated bay.

Custom Interface Required:

The display board uses TTL logic exclusively and supplies the necessary signals for our interface. The interface was designed for us by Warren Stardup, who is a partner-technician with Microcomputer Systems, Inc. of Tampa, Florida.

The interface converts the TTL to serial, using a UART (Universal Asynchronous Receiver Transmitter). Data is then transmitted from the mill site to the office computer via a 20 mA current loop. Each board size is represented by a standard ASCII character. The proper ASCII code is signaled to the UART through a diode matrix on the interface board.

System Configuration:

The system includes an Altair 8800 computer with an upgraded power supply and memory that consists of two Dutronics 8K Static RAM boards and one MITS 4K Dynamic Memory board for a total of 20K. However, only about 10K bytes of this are used for both the BASIC interpreter and the lumber program. Both programs are loaded into the Altair computer through a MITS ACR board.

I hope to have our MITS PROM board with the BASIC bootstrap loader up and running soon, because so far, I'm the only person who can load the computer. On several occasions I've been rousted from bed to go down to the mill during the night shift, because a thunder storm has caused a momentary loss of power. Somehow, these power failures always seem to occur during my "off" shift, usually when I'm at home in bed.

The lumber data comes from the interface to one port of a MITS 2SIO board. This port is wired for a 20 mA current loop. The other port on the 2SIO board is wired RS232 and connects an ADM-3 Lear Seigler

Continued

CRT to the processor. All data is displayed and constantly updated on the CRT. The information shown on the CRT is copied down by hand at the end of each shift. We also have a spare SWTP video terminal and hope to acquire some sort of hard copy printer in the very near future.

Software:

The software was written by Bill Turner, a software genius from Microcomputer Systems, Inc. of Tampa. Bill is also one of the well-known authors of SWTP's BASIC.

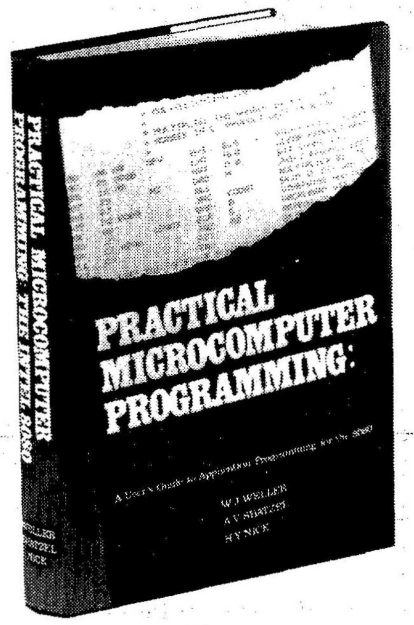
Our program is written in 8K BASIC. Data is displayed in a two dimension matrix (7,6). The display is constantly updated as each board is measured. Boards actually come through the sorter at a rate of one every two seconds, so most of the processor's time is spent simply waiting for an input. To make use of this wasted time, I plan to use another interface which will scan the logs as they are about to be cut and compute the measurement with what comes out of the sorter, to give the actual recovery of all logs that are processed.

The program also has the ability to record all data accumulated on a cassette tape, at a predetermined board count, which is usually every 1,000 boards. In case of a momentary loss of power, this will preserve all of the data accumulated at that particular time. The cassette obviously won't be needed once we get the printer.

Since we began using the Altair system, our inventory discrepancies have been eliminated, much to our bookkeepers' appreciation. But the real merit of the system is that now office personnel can simply glance at the CRT to see exactly what is being produced in the mill. If we have orders for long stock, but the trend on the display is for short stock, we simply get on the intercom and ask our workers to cut the long pieces into shorter ones. This proves invaluable when dealing with truckloads of orders which vary in size from day to day.

8080 programming problems?

IF you need to know how to:



- **service interrupts**
- **do multi-precision arithmetic**
- **convert number bases**
- **handle arrays and tables**
- **control complex peripherals**
- **use the stack pointer**
- **debug your programs**

THEN . . . Practical Microcomputer Programming: The Intel 8080 is the book you've been waiting for. Written by application programming people for application programmers, this is not a book of theory, but a book of step by step solutions to real problems. In eighteen chapters and more than 100 example programs it shows you exactly how to do all of the things listed above and many, many more with 8080 assembly language. A complete programmer's guide to using the 8080, it also contains the full source text of a mini-computer cross assembler and a debug program for 8080 based systems. This could be the best programming information bargain you will ever see.

CN

Northern Technology Books
Box 62, Evanston, Illinois 60204

Please send my copy of Practical Microcomputer Programming: The Intel 8080 at $21.95.

☐ check enclosed ☐ money order enclosed

Illinois residents add $1.10 state sales tax. No C.O.D. please.
Please type or print

Name ____________________
Company ____________________
Address ____________________
City__________ **State**__________ **Zip**__________

CMPCTER-LOADER
Masters Paper-Tape Reader

BY: DOUGLAS L. JONES

The Altair 680b PROM monitor is truly 256 bytes of total cleverness. It enables you to (L)oad, (M)emory inspect and modify, increment to the (N)ext address, (J)ump to a program that is in memory or (P)roceed from the presently interrupted point. That's a lot of software power on a little ROM chip.

If the I/O part of my computer system, an old clunker model-33 Teletype™ were up to snuff, everything would be just fine. But it isn't so. Couple the insatiable ability of my clunker 33 to drop bits on the paper-tape reader with the checks and cross-checks of the (L)oad loop in the PROM monitor, and you've got instant frustration. CMPCTER-LOADER was written in an attempt to show that man is able to master even a paper-tape reader.

The Altair 680b PROM monitor supports what is know as the Motorola paper-tape format. This consists of three important points:

S0 (ignore) (ignore)

S1BBAAAADDDDDDDDCC

S9 (ignore) (ignore)

Where:

S0=Start of a header block which for our purposes can be ignored.

S1=Start of a valid data block.

BB=Block count of all remaining data in this block.

AAAA=Start address to load

DDDD=Hexadecimal data being loaded.

CC=Checksum value.

S9=All data blocks finished (return to exec . . or whatever).

The data blocks are in ASCII and are printable as local copy on the Teletype™. Each is followed by carriage returns and line-feeds. Since they are not part of the data, these characters will not be 'loaded' but will only perform machine control--a well thought out format in all respects.

The PROM monitor easily handles this Motorola Format. The (L)oad command will cause the monitor to begin the load loop routine. Any data that does not begin with 'S1' is totally ignored. Once recognition of an 'S1' data block-mark occurs, the monitor is locked into a data-block format:

1. Only hexadecimal characters are allowed-up to and including the checksum.

2. The checksum is actually computed and must agree mathematically with the data being loaded.

3. An S9 data block will terminate loading.

But herein lies part of the problem. A jump out of the load loop and back to the EXEC of the monitor will occur not only on recognition of an S9 data block, but also in the middle of a valid data block, if it recognizes a non-hex character, or at the end of a data block if the checksum computed is not correct.

Imagine trying to load a rather lengthy paper tape such as the MITS 680 BASIC, with the PROM monitor. The (L)oad command is typed and the paper tape started. One of the first data blocks loaded is address 00F3 with a constant FF. This will at least kill the echo so that your printer does not type out some 480 lines of 'S1datadatadatadata...". But the loading of the tape is rather boring so you decide to run up-stairs to make coffee, or to watch some more of the movie... and that's when it happens.

Inevitably, the paper-tape reader will goof. It might read one of the data characters as a non-hex character and/or it will compute a checksum incorrectly. The PROM monitor says to itself: "Ok. You blew it. The heck with this incorrect jazz, it's back to the EXEC for me."

The program falls through the load loop and returns to the EXEC at entry line CRLF. The EXEC will respond with a (CR) (LF) and then a prompt character (.).

After the prompt character, the only thing that is supposed to be typed is (L), (M), (N), (J) or (P). But the paper-tape reader has meanwhile continued on its merry way. It does not stop shoving non-command characters into the PROM. The PROM is out of the load loop and, therefore, does not like any of the characters that it is receiving. Another character is received, it doesn't 'compute', so it again responds with the prompt characters. Some 42 feet of paper go by on the Teletype™ while you break your neck trying to get back down-stairs to shut it off.

Functionally:

```
10  RECEIVE BUM CHARACTER
20  JUMP OUT OF LOAD LOOP
    AND BACK TO EXEC
30  SEND OUT PROMPT (CR)
    (LF)  (.)
40  RECEIVE A CHARACTER
50  IS IT (L), (J), (M), (N),
    or (P) ??
60  HECK NO (LAUGH A LITTLE)
    THEN JUMP TO 30
```

Relief is on its way. First, by sending either a constant B1 or a D1 to the Teletype™ port (address F000), the paper-tape reader can be under complete software control. That should solve half of the problem--let the computer control the paper-tape reader.

Let's look at something else:

S11300A0BD...

BD is to be loaded at address 00A0. We want to deposit a binary 10111101 at that address. But the 'B' is ASCII 01000010 and the 'D' is ASCII 01000100. Half of each character is information saying that "this is an ASCII character". So it takes a lot of software shennanigans to strip off the ASCII, left justify, inclusive OR the data, etc., just to load BD at one address. If the BD were in binary on the paper tape to begin with, it could be loaded at the appropriate address in half the number of paper-tape characters.

So it looks like we're going to build a LOADER. It will accept pure binary data, so it will cut the load time considerably. It will recognize bummers and will stop the paper-tape reader. These were the starting criteria for the dual programs CMPCTER-LOADER.

HARDWARE REQUIREMENTS

1. 33 ASR Teletype™ running full duplex to the computer. This allows different data to be input and output in addition to suppressing output when needed (killing the echo).

2. Paper-tape reader under software control.

This will allow stopping the reader on recognition of an error.

NOTE: All paper-tape reader control functions must be strapped out. Examples:

EOT End of Tape
X-OFF
WRU Who are you (Kicks of answer-back drum)

Continued

CMPCTER-LOADER
Masters Paper-Tape Reader

Continued

If received, any of these characters will cause something detrimental to happen. Pure binary characters will be sent to the printer, the odds of hitting these control functions are quite high.

Although both the CMPCTER (compacter) program and the LOADER program exist in the Motorola format, they are small enough so that you can manage to get them in even the most finicky Teletype™.

CMPCTER will take your Motorola style 680 BASIC or ASSEMBLER tape and compact it to the new binary format. This needs to be done only once for each of your big tapes.

The loader must be pre-loaded. It's entered before the compacted BASIC or ASSEMBLER tape and is actually the mechanism that brings in the compacted tape. The compacted tape is of a binary nature, so the LOADER will handle it.

CMPCTER

Let's assume that you want to compact a particular program tape--the MITS EDITOR/ASSEMBLER, for example. It's supplied in the lengthy Motorola format. The starting address of the EDITOR is Ø1Ø7. Remember, the tape needs to be compacted only one time.

Using the (L)oad command, load the CMPCTER program via the Altair PROM monitor. During loading, it will kill the echo, stop the tape reader at completion and return echo. Type the letter 'J', so that control returns to the EXEC.

Load the paper tape to be compacted into the reader, but do not start the reader yet.

Start CMPCTER by typing (J)ump ØØØØ.

CMPCTER will respond with a question mark. It's asking you at what address you want to start your compacted tape program.

Type in Ø1Ø7. At the completion of typing the '7', quickly reach over and turn-on the paper-tape punch. CMPCTER will start to punch out the leader.

When the leader stops, turn on the paper-tape reader.

What happens next is rather odd. The computer will read in one block at a time, stop the paper-tape reader and proceed to punch out that block in compacted form. The Teletype™ printer will be making rather weird motions as it's responding to the binary data fed to it. Every so often, some ASCII data might even go by, but usually it will just type gibberish. Let it go. It will take a while, but let it read and punch.

Should CMPCTER pick up either a non-hex character or a checksum error, it will HALT and will not continue either reading or punching data. No problem. Here is how to recover from the error:

Turn OFF the paper-tape punch.
Turn OFF the paper-tape reader.
Reposition the tape in the reader to the middle of the preceding data block.
Hit RESET on the computer and restart CMPCTER at (J)ump ØØØØ.
Turn ON the paper-tape punch.
And finally,
Turn ON the paper-tape reader.
Compacting will then resume.

At the recognition of the 'S9' data block, compacting will cease. CMPCTER will then add its own special data block JUMP mechanism (7E Ø1Ø7) to the compacted tape in order to start your EDITOR program at Ø1Ø7.

This will be followed by a finishing leader. The final thing CMPCTER will do is list how many blocks (in decimal) that it has compacted.

LOADER

LOADER is also in the Motorola format (but it's very short and sweet). Using the (L)oad command, put LOADER into the computer.

Position the compacted tape to be loaded in the reader, and turn the reader ON.

Start the LOADER program at (J)ump 4300.

The LOADER will take over control of the reader and quickly bring in your binary-mode compacted tape.

The LOADER will respond to two types of errors. In both cases, the reader will stop dead in its tracks.

1. It failed to recognize a block-mark correctly.
2. It failed on a computed checksum.

The format of the error messages is included on the assembly listings. To recover from the error:

Back up the tape to the middle of the preceding data block, and turn the reader ON. Block marks are easy to recognize in this compacted form--they are two rubouts in a row (hex FF, FF).

Reset the computer and restart LOADER at (J) 43ØØ. Loading will then resume.

The last data block on the compacted tape is actually a JUMP instruction that will over-write a part of LOADER and cause a jump to address Ø1Ø7. After that, your program will be up and running.

Yes, fixing the paper-tape reader would have solved a lot of problems. As parts become available, it is being repaired. But the added bonus of CMPCTER-LOADER is that the actual load time is cut in half due to the binary mode of the program tape.

About the only critical software item is a constant in CMPCTER, labeled TIME. It appears once (line#ØØ176) at address Ø1EC. This value may need adjusting depending upon how fast your paper-tape reader can be stopped.

Even if you do not own the Altair 680b computer, most microprocessor systems use an ASCII type of tape data format. Writing a CMPCTER-LOADER program in the machine language of your computer will pay big dividends.

Bits & Pieces
Continued from Page Two

NEW FACES HAVE APPEARED IN THE MITS MIRROR

MITS would like to welcome to her ranks Ken Zaike, Charles Olsen III, and David Ning. Ken is now the head of our Industrial Sales Department, Chuck is heading up Public Relations and David now handles International Sales. If you have questions in any of these areas, please feel free to contact these gentlemen.

RENEWAL SUBSCRIPTIONS ARE NOW BEING ACCEPTED

For the cost of $5.00/year ($20.00 to our foreign customers), you will become an Associate Member of our Altair Users Group. As a member you will receive a copy of Computer Notes monthly, plus you will be able to both write and purchase any programs (for a copying charge only) which we have in our Software Library. For payment we will accept the following: BAC, MC, a Bank Check, or a personal check which requires a 3-week holding period. We will be sending out renewal notices in mid January to all those persons whose subscriptions will expire after the January issue. This will allow for the time necessary to submit your renewal order, and not miss any issues of Computer Notes.

Those persons who have already sent in their renewal subscription in the amount of $10.00, will have this money remain on their account until such time that they either request a refund or utilize the money for another order.

If you have any questions, please feel free to contact us.

LOADER

```
$
  NAM LOADER
*
* SOURCE #2.0
*
*  LOADER WILL LOAD IN A PURE BINARY
*  OUTPUT TAPE PRODUCED BY CMPCTER
*  THEN JUMP TO A PREARRANGED ADDRESS
*
* ERROR  MESSAGES
* BMM #### AAAA DD
* CVV #### AAAA DD
*
*     B:  BLOCK MARK ERROR
*     C:  CHECKSUM ERROR
*    MM:  BUM BLOCK MARK
*    VV:  CHECKSUM VALUE
* AAAA:   LAST ADDRESS BLOCK LOADED
*    DD:  CORRESPONDING DATA
* ####:   BLOCK COUNT LOADED
*
*
   OPT P
ACIACS EQU $F000
BYTECT EQU $00F9
BLKCNT EQU $43FE
XHI EQU $00FA
XLOW EQU $00FB
OUTCH EQU $FF81
OUT2H EQU $FF6D
OUTS EQU $FF82
POLCAT EQU $FF24
ACIADA EQU $F001
NCHANG EQU $FF8F
*
  ORG $F3   KILL ECHO
  FCB $FF
*
  ORG $4300
*
BEGIN  JMP RDY
B1 JSR BYTGET
  CMP B #$FF
  BNE B1
  JSR BYTGET
  CMP B #$FF
  BEQ L12   PAST FIRST BLOCK MARK?
  BRA B1
*
  ORG $4316
*
* THE NEXT 3 ADDRESSES ARE
* OVERWRITTEN WITH JUMP INSTRUCTION
LD BSR BYTGET
  NOP
*
*
  CMP B #$FF  RUBOUTS
  BNE KB
 BSR BYTGET
  CMP B #$FF CHECK NEXT ONE TOO
  BNE KB
L12  CLR A  ZERO CHECKSUM
  BSR BYTGET  GET BYTECOUNT
  SUB B #2
  STA B BYTECT  BYTECOUNT
LD11 BSR BYTGET GET HIGH ADDRESS
  STA B XHI STORE IT
  BSR BYTGET  GET LOW ADDRESS
  STA B XLOW  STORE IT
  LDX XHI  LOAD X WITH ADDRESS BUILT
MORE BSR BYTGET  GET DATA BYTE
 DEC BYTECT  DECREMENT BYTE COUNT
  BEQ LD15  DONE WITH THIS BLOCK?
  STA B X  STORE DATA
  INX  BUMP POINTER
  BRA MORE  GO BACK FOR MORE
LD15 INC A  INCREMENT CHECKSUM
LLD  BEQ BUMP  ALL OK
  PSH A
  LDA A #'C  CHECKSUM ERROR
BUM LDA B #$91
  STA B ACIACS  KILL READER
  CLR B
  STA B $F3  ECHO ON
  LDA B #$0D
  JSR OUTCH
  LDA B #$0A
  JSR OUTCH
  TAB
  JSR OUTCH  DUMP TROUBLE TYPE
  PUL A
  JSR OUT2H  DUMP BUMMER
  JSR OUTS  SPACE
 LDA A BLKCNT
  JSR OUT2H  DUMP BLOCK COUNT
  LDA A BLKCNT+1
  JSR OUT2H
  JSR OUTS
* SHOW BUM BLOCK ADDRESS AND RETURN TO PROM
  JMP NCHANG+5
*
KB PSH B
  LDA A #'B  BUM BLOCK MARK
  BRA BUM
*
BUMP CLC
  LDA A BLKCNT+1
  INC A
  DAA
  STA A BLKCNT+'
  LDA A BLKCNT
  ADC A #00
  DAA
  STA A BLKCNT
  BRA LD
*
BYTGET JSR POLCAT  READY YET?
  BCC BYTGET  NOPE
  LDA B ACIADA  GET 8 BIT CHARACTER
  ABA  ADD TO CHECKSUM
 RTS
*
RDY LDA B #$D1
  STA B ACIACS  8 BITS AND READER ON
  LDA B #$FF
  STA B $F3  KILL ECHO
  JMP B1
*
  ORG $43FE  CLEAR BLOCK COUNT
  FCB 0,0
*
  ORG $00F3
  FCB $03
  ORG $F000
  FCB $B1
*
  END
```

CMPCTER

```
$
  NAM CMPCTER
* SOURCE #1.8
*
* COMPACTER WILL TAKE A MOTOROLA PAPER-TAPE
* FORMAT COMPACTING IT TO A BINARY FORMAT FOR LOADER
* IN ADDITION TO ARRANGING A JUMP TO THE PROGRAM
* BEING LOADED
* SHOULD AN ERROR OCCUR, COMPACTING
* WILL BE TERMINATE AND CAN BE RESUMED
*
* START AND RESRT ADDRESS  J 0000
*
*
  OPT NOG
  OPT P
*
TIME EQU $F8  PTRC RELAY CONTROL
*
BYTE EQU $FF53 PROM
BUFEND EQU $000C BUFFER END ADDRESS
BLKCNT EQU $03FE  BLOCK COUNT
BYTECT EQU $00F9 BYTE COUNT
SKELAD EQU $0003 JUMP SKELETON
BUFFER EQU $0010 TEMP BUFFER
BADDR EQU $FF62 PROM
INCH EQU $FF00 PROM
READER EQU $F000 ACIACS
OUTCH EQU $FF81 PROM
OUTS EQU $FF82 PROM
ECHO EQU $00F3
OUT2H EQU $FF6D PROM
TEMP EQU $00F8
STAD EQU $0100
  ORG $F3 KILL ECHO
  FCB $FF
*
  ORG $0000
START JMP BEGIN
SKEL FCB $FF,$FF BLOCK START
  FCB $06 BYTE COUNT
  FCB $43,$16 OVERWRITE ADDRESS
  FCB $7E JUMP INSTRUCTION
JMPADR FCB 0,0  JUMP ADDRESS
SKLCK FCB 0 CHECKSUM
*
  ORG $0100
BEGIN  JSR CH1-1
  JSR CRLF
  LDA B #$3B PROMPT
  JSR OUTCH
  STA B ECHO  TURN ON ECHO
  JSR OUTS
  CLR A  ZERO CHECKSUM
  STA A BLKCNT  ZERO BLOCKCOUNT
  STA A BLKCNT+1
  JSR BADDR  GET JUMP ADDRESS
  STX JMPADR  STORE IN SKELETON
  ADD A #$DD  PRECANNED CHECKSUM VALUES
  COM A
  STA A SKLCK
  JSR CRLF
  CLR B  NULLS
  JSR LEADER
  LDA B #'P  SPECIAL
  JSR LEADER
*
  LDX #*
  STX START+1
LOOP LDA B #$FF
  STA B ECHO  KILL ECHO
LOAD LDX #BUFFER
  LDA B #$FF
  STA B X  START MARKS
  INX
  STA B X
  INX
  JSR CHIN  READ FRAME
  SUB B #'S
  BNE LOAD
 JSR CHIN
  CMP B #'9
  BEQ CONT
  CMP B #'1
  BNE LOAD
  CLR A  ZERO CHECKSUM
  JSR BYTGET GET BYTE COUNT
  STA B X  STORE IN BUFFER
  INX  BUMP
  STA B BYTECT  STORE
LOOP2 JSR BYTGET
  STA B X
  INX
  DEC BYTECT
  BNE LOOP2 DONE?
  INC A INCREMENT CHECKSUM
  BEQ SETDMP ALL OK
*
GOOF BRA GOOF
* ON A GOOF, COMPACTING WILL STOP HERE.
* IT IS CAUSED BY EITHER A NON-HEX CHARACTER
* OR A BUM CHECKSUM.
* BACK UP SOURCE TAPE TO THE BEGINNING
* OF THE STOPPED BLOCK AND RESTART PROGRAM
*
SETDMP DEX  WHOOPS ONE TOO MANY
 STX BUFEND  SAVE END ADDRESS
  LDX #BUFFER  GET START ADDRESS
DMPBUF LDA B X  GET CHARACTER
  JSR OUTCH  DUMP IT
  CPX BUFEND  REACHED END ADD?
  BEQ DONE
  INX
  BRA DMPBUF
DONE CLC BUMP BLOCK COUNT
  LDA A BLKCNT+1
  INC A
  DAA
  STA A BLKCNT+1
  LDA A BLKCNT
  ADC A #00
  DAA
  STA A BLKCNT
  JMP LOOP
*
CONT LDA A #9  LOOP COUNTER
  LDX #SKELAD
C10 LDA B X  DUMP AND BUMP
  JSR OUTCH
  INX
  DEC A
  BNE C10
  CLR B  NULLS
  JSR LEADER
  JSR CRLF
  LDX #STAD  SET TO START NEW TAPE
  STX START+1
  LDA A BLKCNT  LETS DUMP BLOCK COUNT COMPACTED
  JSR OUT2H
  LDA A BLKCNT+1
  JSR OUT2H
*
HALT  BRA *
*
LEADER  LDA A #$32 LOOP COUNTER
L10  JSR OUTCH
  DEC A  DECREMENT LOOP COUNT
  BNE L10 DONE?
  RTS
*
BYTGET BSR HEXGET  GET FIRST HEX DIGIT
  ASL B  SHIFT TO HIGH ORDER 4 BITS
  ASL B
  ASL B
  ASL B
  ABA  ADD TO CHECKSUM
  STA B TEMP  STORE DIGIT
  BSR HEXGET  GET SECOND HEX DIGIT
  ABA  ADD TO CHECKSUM
  ADD B TEMP  COMBINE DIGITS TO GET BYTE
  RTS  RETURN
*
HEXGET JSR CHIN  GET CHARACTER
  SUB B #'0 HEX?
  BMI GOOF NO
  CMP B #$9
  BLE BACK  OK
  CMP B #$11
  BMI GOOF NO
  CMP B #$16
  BGT GOOF
  SUB B #7  ITS A LETTER GET BCD
BACK RTS RETURN
*
WAIT PSH A
  PSH B
 LDA A #TIME
W2 CLR B
W1 INC B
  BNE W1
  INC A
  BNE W2
  PUL B
  PUL A
  RTS
*
CHIN LDA B #$D1  READER ON
  STA B READER
  BSR WAIT
  JSR INCH
  PPH B
CH1 LDA B #$B1  READER OFF
  STA B READER
  BSR WAIT
  PUL B
  RTS
*
CRLF LDA B #$0D
  JSR OUTCH
  LDA B #$0A
  JSR OUTCH
  CLR B
  COM B
  JSR OUTCH
  JSR OUTCH
  RTS
*
  ORG $00F3
  FCB $03
*
  ORG $F000
  FCB $B1
  END
```

input signal (SINP) is of importance--during M3 of the input instruction, SINP goes high enabling circuitry that allows data from the I/O card to be input during T3. SINP goes low again at the next coincidence reflecting a different instruction. To further set the stage, during each of the machine cycles the processor outputs an address (either for memory or I/O) which is presented less than 300 nsec. after the start of that machine cycle or about halfway through T1. The problem occurs during T1 of the RRC instruction, because the coincidence of SINP and the address for that instruction (which happens to be the data channel address) resets the "data available" FF in IC M--pin 19 on the I/O card. As written, the program will then cause a reset pulse of 200 nsec. at pin 8 IC G (SIO-B) during T1 of RRC every time through the loop. If the "data available" FF is set after the status word is input from IC M during T3 of M3 but before the reset during T1, then the indication of a character present is lost as well as the character itself. Normally, the reset doesn't occur until the input instruction at location 007 is executed with the data channel address. In that case the character is input and the "data available" FF is cleared to indicate that fact.

This situation will occur with the use of the SIO cards and the ACR and possibly with other I/O cards. While a hardware solution is not crucial to the operation of a system, it could be unwieldy to use interrupts or write programs in specific locations, particularly if the number of I/O cards were increased.

Sincerely yours,
Jim Wiggins

Altair Users

Don Chamberlain
9457 Las Vegas Blvd. South #321
Las Vegas, Nevada 89119
Phone: (702) 361-4924

Alfred R. Howes
Box 342 Boyce Rd.
Glenford, NY 12433

Dick Fehriback
5779 Blaine SE
Grand Rapids, MI 49508
(616) 455-3138

HAM on the side

By Dave LeJeune

David LeJeune is a Lieutenant Colonel in the Army Signal Corps. He graciously agreed to take over HAM ON THE SIDE.

Ignoring a long-standing policy that I picked up early in my Army career to never volunteer for anything, I now find myself with a monthly column to write about the marriage of amateur radio and computers. My appeal goes out early--HELP! From the great amount of chatter I hear on the ham RTTY nets, the C.W. bands and the voice frequencies, I assume that there are a lot of hams out there playing around with microprocessors. This column offers all you enthusiasts the perfect opportunity for letting everyone else know what you're doing. The success of this column depends upon reader acceptance, and that's measured by the amount of feedback (good or bad) you give me. So start sending those cards and letters now!

I've been a ham for about 20 years, so I'm old enough to remember the changes caused by the introduction of single side band(SSB) to our ranks. SSB completely revolutionized HF voice operating. But, as often happens with many new products, people didn't fully accept SSB at first and simply termed it a new fangled device. Within 10 years, however, it had become the dominant mode for voice communication. In the mid 60's the re-introduction of FM (it had been used by many for a short time immediately following WWII) had the same effect on the VHF bands.

The use of repeaters caused this mode of operation to spread like mad, and today, most cities where more than two or three hams reside have some sort of VHF repeater available.

The introduction of the microprocessor had the same effect on amateur radio as the introduction of SSB and FM repeater techniques. The microprocessor can be used for such applications as RTTY and C.W. as well as station control, logging and contest operating. However, with a little help from the FCC, the microprocessor may also find its way into voice operating.

In the commercial communications field the trend is towards a total digital network. With such a network it's easier to multiplex many digital channels onto a single transmission system than to analog channels. Designing circuit switches is also easier. But many problems have yet to be solved.

For instance, if voice recognition is to be maintained, a digital voice channel must use twice the channel bandwidth as a comparable analog channel. The channels should require about the same bandwidth for a given power level and signal-to-noise ratio. Other areas in the development of optimum error detecting and correcting codes are still wide open to exploration and experimentation. The only tools needed to attack such problems are an amateur radio station with some means of generating the necessary digital voice, RTTY or C.W. signals, a microcomputer and some means of interfacing these two systems.

As mentioned previously, the FCC needs to help out by easing current regulatory restrictions imposed on amateur radio operators. As a starter the restriction of RTTY operation to the Baudot code should be removed. This would bring the RTTY ham out of the dark ages and into the world of computer communications. Secondly, in order to attack the problems of digital voice transmission and error correcting codes, the restrictions prohibiting the transmission of codes and cyphers should be removed. The FCC has recently released a new proposal(the "bandwidth" proposal) which will help lift these and other restrictions that currently limit our ability to experiment over the air. However, the proposal seems to be hung up somewhere within the FCC. A note to your congressman might just be the thing to spring it loose.

I would like to discuss these and other topics of interest to the ham/computer freaks in future issues of C.N. Interested readers can once again communicate with me via the MAILBOX system(channel 14.075000 mhz) described in last month's C.N.(It went off the air temporarily in November while the system was being upgraded to an Altair 8800b.) Or, you can just as easily drop me a note in care of C.N. to let me know what sort of information you'd like to see in this column.

SMALL COMPUTER NETWORKING — BIG STUFF

By: Bob Matthews

Two news items from the October-November issues of Datamation are indicative of two major, but seemingly opposite, trends in computing: the move toward bigger, more efficient computers and the trend toward smaller, more efficient computers.

In one of the news items (Datamation, Oct. 1976, p. 17), the Los Alamos Scientific Laboratory in New Mexico announced the delivery of its new supercomputer, which can execute over 20 million floating point operations in one second! (But who's counting?) Obviously, the potential capabilities of such supercomputers are still being explored, and more such developments are certainly forthcoming in New Mexico as well as other states.

The second news item (Datamation, Nov. 1976, p. 17), notes that a bank in Boston has cancelled its order for several moderately large computer systems. The cancellation was due not to lack of money, but the bank's decision to convert its original order into an even larger order for 200 minicomputers. The bank figured it can accomplish things more efficiently and economically with a system of small computers rather than with a few large computers. The bank says that within three years there won't be anything larger than a PDP 11/70 in the house.

The bank in Boston isn't the only institution that has opted to disperse its computing power throughout the organization. Business, scientific laboratories and educational institutions are also tying their computers together in networks to allow small computers, working in tandem, to do the work of big computers. (See Digital Design, Feb. 1976, pp. 26-27; IEEE Spectrum, July 1976, pp. 91-93; Business Week, July 5, 1976, pp. 38-44.)

A few examples will show some of the capabilities of distributed processing in the hands of a small system user.

A. A Minimal System. User A is a graduate student in astrophysics and is doing a research in stellar structure. This research requires the solution of some fairly complex differential and integral equations, most of which have no analytic solutions. The numerical solutions to these equations require huge amounts of raw computing power but relatively little input/output sophistication. After crunching numbers for several minutes or even hours, the computer might only print a few pages of numbers and a graph or two. Since grant money is limited these days, especially for theoretical astrophysics, User A needs a low-cost system that has modest I/O facilities, program storage and editing facilities and enormous floating point mathematical power. Until recently, no system combined all of these capabilities. But now with the age of distributed processing, a system for User A might look like this:

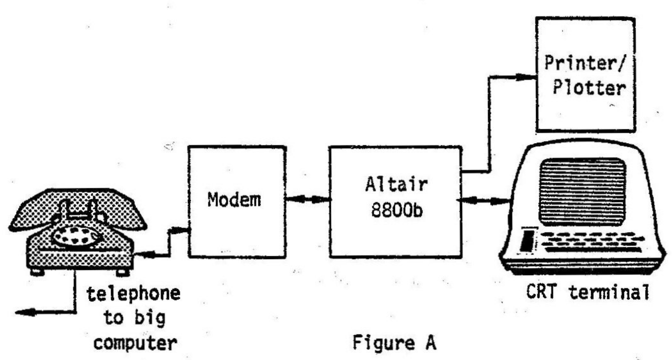


Figure A

The heart of the system is an Altair 8800b microcomputer. It controls a CRT terminal, a printer, small plotter and a floppy disk unit. It is connected through a dial-up or private line to the university's (or whoever's) computer center. A's big simulation program, which A's supervisor and his colleagues pieced together over a period of years and to which A has been making modifications all along, is stored on a floppy disk. The program can be loaded into the supercomputer through the telephone line. To make any necessary changes, A can use the text editor, which runs on the Altair computer, and either save the resulting code on disk or send it to the big computer to be executed. Once the program is loaded, A can go home (or more likely, to the library) and let his Altair system handle the housekeeping details. It can log intermediate output, error messages and even provide input for the big program when it is needed. Finally, it can receive the output of the program and store it for later recall. When A comes in, the output can be examined or "massaged" to make it easier to handle, or the error message file can be examined to see what went wrong.

This is a brief example of a simple system, but it illustrates how the intelligence of an Altair 8800b could be used to help A's research along by allowing more time for studying instead of running back and forth to the computer center. Such a system would allow more flexibility in changing the program and data around to fit the requirements of the project. It would also create more space on A's desk, which otherwise would be overflowing with reams of printer paper and card boxes.

A's system has plenty of speed to handle the CRT terminal and the phone line simultaneously, but there isn't much time left over to do computing on the data as it is being shifted around. If more computing time is necessary, and A, for example, wants to change the format of the data or apply some relatively simple calculations before it is sent to the big computer, one of two things must be done. Either the software must be written to allow interleaving the tasks of I/O and computation, or the following modification must be made to the hardware.

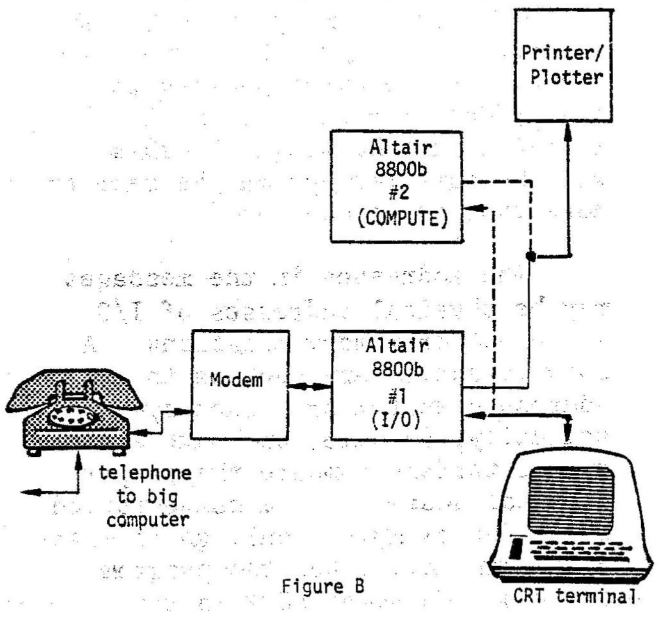


Figure B

Expanding the System.

B's system has another Altair 8800b added whose only duty is computation. This leaves the first computer the task of input and output.

Alternately, the second computer could be a stand-alone system, controlling the terminal and disk while the other is handling communications chores for the big program.

Figure C shows a configuration of a five-computer ring network. The network consists of a complex of small computers surrounded by associated peripheral equipment. All of these "Minisystems" communicate with each other through a unidirectional ring. Such a ring network obviously has tremendous potential for sharing the resources of small computers.

Continued

The University of California at Irvine is just one of many organizations throughout the country which is just beginning to realize that full potential with their own experimental ring network system. (See Datamation, Feb. 1975, pp. 45-46). Their system is similar to the one shown in Figure C.

Each processor in the ring is autonomous. It needs none of the other systems in the ring to operate. However, the ring structure allows the computers to share tasks and to break up large problems into smaller blocks which can be executed in parallel.

The ring works like this: When one of the computers in the system needs assistance--either access to a program in one of the other computers or data in one of the other memories, it sends a message by way of the ring. A message could be a request for service, a packet of data or a series of commands. Messages flow around the ring in one direction. At each ring interface the address on the message is compared with a table of addresses in the computer's memory. The interface either copies the message into its processor's memory and sends it along on the ring or just sends the message on without copying it. When the message gets around the ring to the processor that sent it, it is taken off the ring. In this way the same message may be sent to more than one processor.

The addresses in the messages may be physical addresses of I/O equipment or memory locations. A more versatile approach is to address computer processes symbolically, by name, and have each ring interface compute the proper physical address. A message marked EDIT, for example, would go to whatever computer(s) has the program or data file named EDIT in its memory.

Since the ring is unidirectional, timing is no real problem, because the ring interfaces just listen until the ring is not busy and is able to put on a message. This means the ring may be expanded indefinitely, as long as the physical facilities can handle enough information.

In fact, one of the systems in the ring might be A's or B's system, which would make the resources of the "number crusher" available to all users of the system.

As might be guessed, the hardware problems of these approaches to distributed processing are not excessive--the technology is readily available and not terribly expensive. However, software is a big problem for several reasons.

The I/O routines must be written to accommodate the communications protocols of the system-message format, addressing conventions, hardware limitations, etc. File maintenance routines must handle the formidable problems of file protection in a system where all the users have access to all the files. In order to take advantage of the potential reliability of a distributed processing network, the operating system should be written so that it can be executed on any of the computers in the system, in case one of them fails. (Murphy's law states that the processor that fails must be the one with the operating system.) Reliability would increase if parts of the operating system could be executed simultaneously in several processors, as is the case in the UCI system.

The user or group of users that tackles these problems certainly has his work cut out for him. But one consolation is that such users will be doing work on the frontiers of information science that will become increasingly valuable as the distributed processing revolution spreads. Another consolation is imagining the power and convenience that will be at their fingertips when they finish.

Of course this article doesn't come close to exploring the whole field of distributed processing. Some exciting results are coming out of experiments with shared memory, heierarchical networks and parallel processor structures. Those interested should check the literature. Reports of C.N. readers experiments in this area are invited and will be distributed for other Altair system users to process.

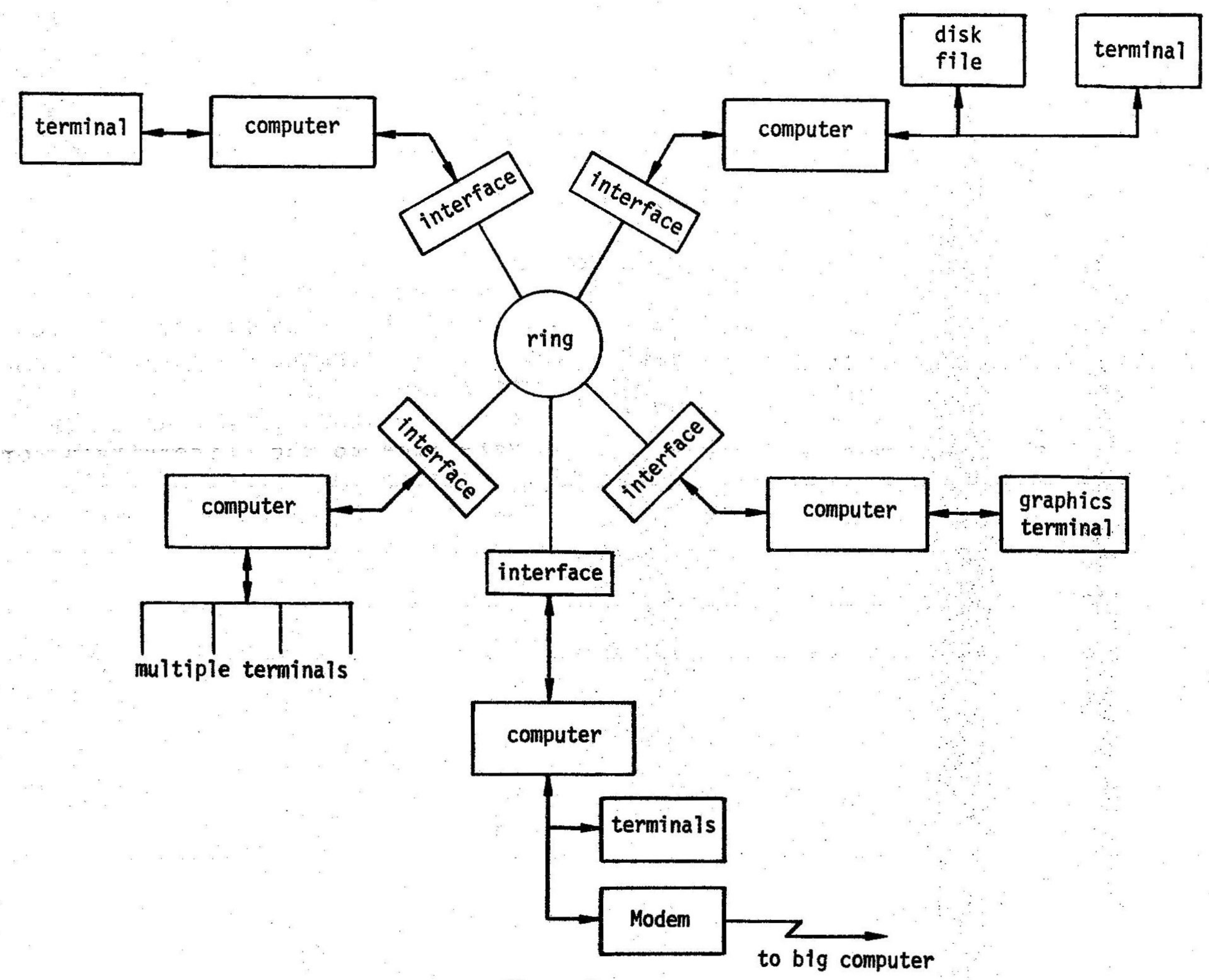


Figure C

New Prices

Two new products have been added to the Altair price list. These prices are now in effect:

	Kit	Assembled
Altair 680b Universal I/O board		
- with no options:	$110	$160
- with one serial port, add:	$ 55	$100
- with two parallel ports add:	$ 50	$ 75
- full board (one serial port and two parallel ports):	$215	$335
Altair 88-Process Control Interface board		
- Assembled Only:		$235

Have you written Software for your Altair™ Computer?

The Altair 8800 computer was the first micro produced for the general public and remains number one in sales, with more than 8,000 mainframes in the field. The wide acceptance of the Altair computer and its rapid adaptation to many diversified applications has truly turned the dream of the affordable computer into a reality.

Yet the machine itself, remarkable as it is, represents only the beginning. The right Software, tailored to meet a user's specific requirements, is a vital part of any computer system. MITS wants to insure that Altair users everywhere have the best applications software available today and in the future. For this reason, a new MITS subsidiary, the ALTAIR SOFTWARE DISTRIBUTION COMPANY, has been formed. Its purpose: to acquire the highest quality software possible and distribute it nationally through Altair Computer Centers.

That's where you come in. The ASDC will pay substantial royalties to the originators of all software accepted into the ASDC library. If you have written business, industrial or commercial use software for the Altair 8800, ASDC wants to hear from you. It is the aim of the ASDC to stimulate and reward creativity in producing useful software that makes those dreams of "computers for everyone" come true. The ASDC will select only software that measures up to its high standards for system design, coding and documentation. The software will then be further documented and distributed through Altair Computer Centers around the country.

For more information on how to submit software to the ASDC, ask your Local Altair Computer Center for an ASDC Software Submittal Packet or contact the ALTAIR SOFTWARE DISTRIBUTION COMPANY.

A subsidiary of MITS

ASDC

ALTAIR SOFTWARE DISTRIBUTION COMPANY
3330 Peachtree Road, Suite 343 Atlanta, Georgia 30326 404-231-2308

see next page for a listing of Altair Computer Centers

ALTAIR COMPUTER CENTERS

BEAVERTON, OR 97005
8105 SW Nimbus Ave.
(503)-644-2314

BERKELEY, CA 94710
1044 University Ave.
(415)-845-5300

SANTA MONICA, CA 90401
820 Broadway
(213)-451-0713

DENVER, CO 80211
2839 W. 44th Ave.
(303)-458-5444

ALBUQUERQUE, NM 87110
3120 San Mateo N.E.
(505)-883-8282;883-8283

TUCSON, AZ 85711
4941 East 29th St.
(602)-748-7363

LINCOLN, NB 68503
611 N. 27th St.
Suite 9
(402)-747-2800

LITTLE ROCK, AR 72206
2412 Broadway
(501)-371-0449

TULSA, OK 74135
5345 East Forty First St.
110 The Annex
(918)-664-4564

HOUSTON, TX 77036
5750 Bintliff Drive
(713)-780-8981

RICHMOND, VA 23230
4503 West Broad St.
(804)-335-5773

SPRINGFIELD, VA 22150
6605A Backlick Rd.
(703)-569-1110

CHARLESTON, W. VA. 25301
Municipal Parking Building
Suite 5
(304)-345-1360

EAGAN, MN 55122
3938 Beau D'Rue Drive
(612)-452-2567

ANN ARBOR, MI 48104
310 East Washington Street
(313)-995-7616

WINDSOR LOCKS, CT 06096
63 South Main Street
(203)-627-0188

PARK RIDGE, IL 60068
517 Talcott Rd.
(312)-823-2388

ST. LOUIS, MO 63130
8123-25 Page Blvd.
(314)-427-6116

NASHVILLE, TN 37203
1600 Hayes St.
Suite 103
(615)-329-1979

BURLINGTON, MA 01803
120 Cambridge St.
(617)-272-8770

ALBANY, NY 12211
269 Osborne Road
(518)-458-6140

NEW YORK, NY 10018
55 West 39th St.
(212)-221-1404

ATLANTA, GA 30305
3330 Piedmont Road
(404)-231-1691

TAMPA, FL 33614
5405 B Southern Comfort Blvd.
(813)-886-9890

ASDC
3330 Peachtree Road, Suite 343
Atlanta, Georgia 30326

altair T.M.

Number One in low-cost computing.

Altair, from Mits, is the number one name in microcomputers for home, business, personal and industrial applications. Because the Altair was first, it has set the standard in the industry. More Altair 8800's are now operational than all other microcomputers combined.

Whether you buy a $395 complete computer kit* or a multi-disk system for under $10,000; Mits will provide you with thorough and lasting support. Satisfied Altair users include schools, corporations, small businesses, students, engineers, and hobbyists.

Altair hardware includes three microcomputers; the Altair 8800a, 8800b, and 680b. Mits has a complete selection of Altair plug-compatible memory and interface options, including the new Altair 16K Static board and Altair multi-port serial and parallel I/O boards. Also available is a complete line of Altair peripherals including line printers, CRT's, and multiple disk systems.

Altair software is by far the most complete and best for any microcomputer. Our Extended BASIC and Disk BASIC have received industry wide acclaim for programming power and efficiency. Application packages are available at many Altair Computer Centers.**

The Altair computer is a revolution in low cost computing. Shouldn't you write for more information including our free, color catalogue.

*The Altair 680b turnkey model.
**Retail Altair computer outlets now opened in many large cities.

mits

MITS, Inc. 2450 Alamo S.E./Albuquerque, New Mexico 87106

Correction

The following chart is a revised edition of the one that appeared in the "8800 Software Tidbits" article by Mark Chamberlin in our November issue. Notice that the numbers listed under "Terminal SS Up" have been increased by one.

Device Type	Octal Code	Terminal SS Up	Load Device SS Up
2SIO (2 stop bits)	∅	none	none
2SIO (1 stop bit)	1	A12	A8
SIOA,B,C (REV 1)	2	A13	A9
ACR	3	A12,A13	A8,A9
4PIO	4	A14	A1∅
88PIO	5	A12,A14	A8,A10
High Speed Reader	6	A13,A14	A9,A10
Terminal at Non-Standard Address	16	A13,A14,A15	Not Supported

Q&A Hardware GLITCHES

By Rich Haber and Bruce Fowler

Our new column, "Glitches," will appear regularly in C.N. and will be devoted entirely to answering hardware questions submitted by our readers. Any type of hardware question(s) is welcome, and we'll try to answer as many as space permits each month. For this first column we've chosen four topics frequently asked about in the past.

1. Static/Dynamic Memory Boards

There still seems to be some confusion, particularly among those who are new to computers, about the differences between our static and Dynamic Memory Boards. Hopefully the following brief explanation of the functions as well as the advantages of each board will clear up any confusion.

Our 4K Dynamic Memory Board uses NMOS technology and has the features of high speed and low power consumption. Each memory cell contains capacitive elements which help to latch the data bits. These cells must be recharged about every 2 milliseconds, or data will be lost. This recharge or "refresh" cycle is accomplished by counting up on address lines A0-A5. The chips are designed so that reading one address refreshes all the elements on one row. There is one row for each of the 64 combinations of A0-A5. During this refresh cycle, the board causes the CPU to execute a "wait" state so that the CPU doesn't attempt to read data from memory while the refresh process is in progress.

This refresh procedure and the other housekeeping circuits related to it require quite a bit of extra circuitry on the board. However, the dynamic design of the board makes it useful where low power consumption is important.

The 4K Synchronous is our new dynamic board which uses existing timing pulses to synchronize its operation. This eliminates the single shots and the need to "tune" the board. It also eliminates the "wait" state, because refresh occurs when the CPU is taking care of its own business.

Compared to Dynamic Boards, the 4K Static Board has a relatively simple design. The 4K Static Board was designed with the 2102 1Kx1 chip, which was the best chip available at the time. It's still an excellent memory chip, but it has only one-fourth the storage capability of the 4060 dynamic chip. So, 32 ICs were required to do the job of eight. Even with the additional circuitry required for address decoding, the board still has a very simple design. The individual cells behave as stable switches and don't need to be refreshed.

The disadvantages of the board are that it's somewhat slower and draws over twice as much current as the Dynamic Board. The advantages are low cost, high reliability and no "wait" states.

The 16K Static Boards are extremely reliable storage devices. They use the more sophisticated 4200 4Kx1 memory chip. The board is designed so that the chips are not selected, except when they are read from or written into. The result is that the board requires the least power and has the fastest access time of any MITS board.

2. Sense Switches

We still get many questions about sense switches and how to use them. The sense switches (Switches A8-A15 on the front panel) represent a byte of data that software can read and use to select options.

These switches are hardwired to input port 377 (Octal). There is no control address associated with sense switches, and output to sense switches is not meaningful.

However, by doing an input from address 377 (octal), the current condition of switches A8-A15 will be read as one byte of data. A switch in the up position represents a one. By software manipulation, the condition of any particular bit or switch can be tested. A branch to a different part of the program can be made, depending on the state of the switch.

For example, before Altair BASIC responds with "MEMORY SIZE?" on a terminal, it checks the sense switches to see what input-output board is being used. BASIC will do an input on address 377 (octal). Depending upon which sense switches are up, BASIC will jump to a specific address in memory. The instructions at this address will then tell it which I/O board to respond on. Check page 97 of the BASIC Manual to see which sense switches correspond to which boards.

A byte from sense switches can be input by using machine language (333 377) or by using BASIC (INP (255)).

In terms of troubleshooting, sense switches can be quite useful. For example, the ASCII equivalent of a letter can be put on switches A8-A15. By inputting sense switches and then outputting on a terminal input-output data channel, the output function of an input-output board can be checked. For example, for a 2SIO at channel 20 (octal) with A8 and A14 up, the letter A should be printed continuously when running program 1, listed below:

Program 1:

```
076
003
323
020
076
021
323
020   Program outputs on data
333   address 021 (for 2SIO)
020   a character whose ASCII
017   code is on the sense switches.
322   The first 8 bytes initialize
010   the 2 SIO's ACIA.
000
333
377
323
021
303
010
000
```

A similar program in BASIC that prints out any changes in sense switches is given on page 30 of the Altair BASIC Manual.

The Repair department uses program 2, listed below to test IC U and T on the 8800A front panel. Data input from the sense switches is stored in location 000 040. If all ones are stored in location 040, no matter which sense switches are up, it usually indicates a bad IC U and T.

Program 2:

```
333
377
062   Program 2 stores sense
040   switch pattern in location
000   000 040.
303
000
000
```

Perhaps the most powerful aspect of sense switches is the ease at which they allow the programmer to input data to the computer while it is running programs. If software monitors the sense switches, different programs can be selected and executed depending upon which sense switches are up.

3. Interchanging TTL Chips

Customer inquiries about the possibility of interchanging TTL chips are quite common in the repair department. Most of the questions come from customers who say that they cannot purchase replacements locally but still hope to get their Altair computers up and running by making substitutions. For example, replacing a 74123 with a 74L123 is an often asked about substitution. However, this and most other exchanges are undesirable and some can be destructive.

Continued

Glitches
by Rich Haber & Bruce Fowler

Continued

Whatever chip comes with the system has been chosen for its particular characteristics and should not be replaced with any other chip. The two tables below, which give some data on the most important characteristics of the different 7400 logic families, should make this concept more clear and also provide an explanation of the relationship between logic families.

TTL Sub-families
7400 - standard TTL
74H00 - High power TTL
74L00 - Low Power TTL
74S00 - Schottky TTL
74LS00 - Low Power Schottky TTL

Speed and fanout are the most important considerations when dealing with replacement of an individual IC. Changing the speed of a device can have a significant effect on a system timing. Low speed devices can often be used as digital filters, because they cannot respond to certain "glitches" that are on a signal.

Fanout, a factor of input and output currents, can also be very critical. For example, a 7400 can drive about 40 74L00 inputs, but a 74L00 can drive only two 7400 inputs. This could cause not only immediate system failure, but possible permanent damage. For example, if the lack of drive caused a memory board not designed for permanent selection to always be selected, possible damage could result to the regulator. Of course, any such alteration would probably void your warranty.

Another problem arises when attempting to substitute timing ICs, such as one-shot multi-vibrators. The timing components are different for not only each of the families of the same type (i.e. 74123 and 74L123), but also for the same type by different manufacturers!

So, to save yourself some head-aches, we suggest the use of exact replacement parts. If they are unavailable at your local Altair dealer, check with us--we usually carry a full stock of all parts.

RELATIVE CHARACTERISTICS

	7400	74H00	74L00	74S00	74LS00	Units
Supply Current (Average per Gate)	1.0	2.5	.11	2.5	.2	ma
Typ. Turn on time	7	6.2	31	2	31	ns
Typ. Turn off time	11	5.9	35	3	9	ns

FANOUT CAPABILITIES

	7400	74H00	74L00	74S00	74LS00
7400	10	8	40	20	8
74H00	12	10	50	25	10
74L00	2	1	20	10	1
74LS00	5	4	40	20	4
74S00	12	10	100	50	10

(The families at left will drive the listed number of loads tabulated at right.) Data is from the National Semiconductor TTL Data Book. C. 1976, National Semiconductor Corporation.

Articles Welcome

Computer Notes is continually seeking quality manuscripts on applications, troubleshooting, interfacing, software, book reviews, fiction and a variety of other computer-related topics. Articles and article ideas are always welcome.

Payment for articles which are accepted will be sent within 4 weeks after receipt of a manuscript. Honorariums for articles are based on technical quality, writing ability and suitability for C.N.'s readership, and range from $10 up to $50 per typeset magazine page. All articles are subject to editing to fit space requirements and content needs of our readership.

Articles submitted to C.N. should be typed, double-space, with the author's name, address and the date typed in the upper left-hand corner of each numbered page. Authors should also include a brief statement about their job title, background and/or experience written separately under the article's title.

Photos, charts, programs and figures should be clearly labeled and referred to by number within the text of the manuscript. Only clear, glossy black photos (no Poleroid pictures) can be accepted. All photos should be taken with uniform lighting and sharp focus.

Program listings should be recorded with the darkest ribbon possible on new blank white paper (not thin paper).

Articles which are not accepted will be returned.

classified ads

Memory For Sale: GE Core Memory 16K by 40 bits, all power and interface circuitry and documentation--been used with an 8080 CPU. Litton Industries Drum memory with full read/write electronics. Bryant MBM drum memory. By unit or best offer over $250 for lot. Contact:

C. H. Eby
7101 Mammoth Ave.
Van Nuys, CA 91405
(213) 988-1763

Help! I'm in great need of an SG-1900 Audio Sweep Generator. The company where I'm employed purchased the SG-1900 and based a large part of the company business around it. But, unfortunately the system was stolen while it was being displayed at a computer show.

The company is willing to purchase, pending previous inspection, any new or used SG-1900. Please drop me a note if you have any information about the sale of an Audio Sweep generator.

Larry Nelson
344 Quince St.
Apt. #1
Salt Lake City, Utah 84103
Tel: (801) 521-2540

FIELD TESTING THE ALTAIR 8800

By Gene Dial

Dr. Dial is a professor in the Graduate School of Public Affairs at the University of Colorado. This article first appeared in the September DACS Newsletter.

A short time after completing the assembly of my Altair 8800 computer and brushing up on BASIC, I began to look for an opportunity to field test it for reliability and accuracy. This opportunity presented itself with the Boulder Country Club Frolics--an annual golfing event in which members teamed with guests from around the country in achieving the golfing and social highlight of the year.

Scoring presented a problem in the manual mode. Points were to be awarded on each hole by comparing strokes required with the par value of the hole, and then by adjusting the gross count by a handicap-derived "quota". After the first day of play, the quota was adjusted, plus or minus, by 50% of the points earned the first day. Extending 128 players by 18 holes, three days of play, and two quota settings indicated a need for almost 14,000 calculations based on about 7,000 data entries.

In the past this chore was performed each evening by a committee of three with the help of an adding machine. Accuracy required agreement between three independent evaluations of each score card, or some 41,000 calculations in the manual mode, plus redundant checking to resolve divergent evaluations of the same card. Obviously, the club welcomed relief from this formidable task. But it was also wary of being screwed by the "blue box". Reliability was more at issue than time.

I devised a user-oriented program in which the layman-operator was led through each sequence by simple questions presented on the screen, e.g., "DO YOU WANT TO INPUT DAY-1 SCORES?", to which the operator would respond with a "Y" or "N". Having made a choice, the screen would then provide the instructions to implement that choice.

The program wound its way eventually to the "Display Module". In this mode, a series of displays was selected which depicted names of players, the handicaps of each, the initial quota of each team, 1st day scores and points, revised quotas, 2nd day scores and points, finals, and Bulletins. All displays were framed with appropriate heads, were indexed so as to achieve a page-to-page effect, and were provided with an appropriate pause to permit casual reading of each page.

I had assembled a television interface kit (PIXIE-VERTER #PXV-2A, available at Gateway's for about $10), and with antenna line splitters, connected the computer to the club antenna system. This produced a commercial quality signal on channel three to each of the club's TV sets in several locations.

The display module sequenced through its parts then looped to the beginning for as many loops as the operator requested. The times required for each kind of loop were presented on the screen so that the operator could correctly judge the required number of loops before scores were scheduled to be received. Such calculations were important since the computer was intended to be in the RUN mode for the three-day period.

By way of preparation, members of the Frolic's Committee were given an orientation in a trial run of the program. And, more importantly, the operator, the staff executive of the committee, received some 10-hours of practice over a three-week period.

The equipment was installed and checked out a day ahead of time. Since a quiet, secure location was desired, it ended up in a remote exercise room serviced by an extension cord from another room. I was quite sure that cord would be pulled during the three-day period and memory would be lost, thus indicating a need for adjustment to my plans.

The machine performed well in the first hour only to produce garbage in the second. The reason became evident by the fact that I was sweating. By opening an exterior door and installing two large fans, the Altair and I returned to normal operation.

Scores were scheduled to be received by the morning players at 1 p.m., and by the evening group at 5:30, each group to arrive at one time since a shot-gun start was being employed. Scores were to be inputted on-line in the RUN mode. The screen would ask the team number, then for player number one to announce the hole number and ask for an input. After scores for 18 holes had been provided, the same sequence would be repeated for player number two of the same team. The screen would then ask if there were more scores, and so on through the end of scoring.

Unfortunately, the operator became so proficient that he soon anticipated the sequence and outraced the computer. This resulted in being kicked out of RUN mode and very often an inability to return to it without loss of data. After repeated inputs of the same scores and the same results, I decided to make a major revision in the program. Such circumstances--surrounded by two anxious golf pro's--are not the best for programming. However, it was soon accomplished.

In the new mode, scores were to be entered as program data statements. This permitted casual editing. After each group of scores was entered, a tape was made of the program, thus providing back-up for the memory loss contingency.

At least, this is the way I planned it. Implementation, however, presented new problems. I had exceeded my 24K of available memory. Thoughts of rushing to Gateway to borrow a 4K board were sequenced by a plan to divide the program into two programs: a Scores Input and Preliminary Processing Program; and a Displays Program.

Two anxious golf pro's were now joined by members of the Frolics Committee as I began to disect the program and to ferret out the UNDEFINED GOTO's, NEXT WITHOUT FOR's, and RETURN's WITHOUT GOSUB's. I can't recall that my programming ever got so much attention.

In an hour or so it was done and the new procedures, including provision for back-up recordings, were implemented. Smiles all around, sighs of relief, and the room thinned out. It was going well. Now, I could use some company.

During the remainder of the three days, each golf pro checked out on the system and became expert at inputting and processing scores; the turn-around from receipt of first score to reporting results in Display Mode on video was reduced to 45 minutes; and, I was plied with free booze (hour after hour) and food by a much relieved Frolics Committee.

The programs are now preserved on tape and hard copy, notes have been set down on procedures, and all-in-all, I look forward to a free ride next year.

One final note, the computer emerged without error in 19 score challenges presented during the three days. In each case, other factors--the golfer's ability to calculate, most prominently--were at fault. The Altair reliability and accuracy index rose significantly in my mind as a result of this field trial.